UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/725,016	12/02/2003	Serguei M. Beloussov	2230.0020000/MBR/GSB	3183

54089 7590 05/14/2008

BARDMESSER LAW GROUP, P.C.
910 17TH STREET, N.W.
SUITE 800
WASHINGTON, DC 20006

EXAMINER	
NGUYEN, PHILLIP H	

ART UNIT	PAPER NUMBER
2191	

MAIL DATE	DELIVERY MODE
05/14/2008	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	Application No.	Applicant(s)	
	10/725,016	BELOUSSOV ET AL.	
	Examiner	Art Unit	
	Phillip H. Nguyen	2191	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE *3* MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *14 February 2008*.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-22,24-38,40-50,52-57,59-64 and 67-73* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-22,24-38,40-50,52-57,59-64 and 67-73* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

DETAILED ACTION

1. This action is in response to the amendment filed 2/14/2008.

2. Claims 1-22, 24-38, 40-50, 52-57, 59-64, and 67-73.

Response to Amendment

3. Per applicant's request, claims 1, 9, 24-27, 29-31, 33, 40, 46, 53, and 67-71; Claims 23, 39, 51, 58, 65, and 66 have been amended.

4. The rejection to claims 67-68 under 35 U.S.C. 101 of previous action is withdrawn in view of applicant's amendment to replace recording medium with removable storage unit.

5. Claims 9-12, 31-34, 44-47, 57, 59, and 60 were indicated allowable in the previous action are withdrawn for further search and consideration.

Response to Arguments

6. Applicant's arguments with respect to claims 1-22, 24-39, 40-50, 52-57, 59-64, and 67-73 have been considered but are moot in view of the new ground(s) of rejection.

Claim Rejections - 35 USC § 102

7. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

8. Claims 9 and 31 are rejected under 35 U.S.C. 102(b) as being anticipated by Mulchandani et al. (USPN 6,112,025).

As per claims 9 and 31:

Mulchandani further teaches:

identifying original instructions to be changed while the original instructions are being executed on a processor (see at least col. 5:6-14 "*The relocation type of each patch instruction defines which bits of the instruction to be patched (i.e.* ***dynamically modified during execution of the program****) are provided by original object code instruction and which bits need to be replaced by the address of a constant pool entry. Furthermore, the object code array 252 in the initial compiled file 250 includes the object code for the instructions that will be* ***dynamically modified during program executing***");

copying the original instructions to a storage location (see at least col. 5:18-52 "*the native method loader 216* ***copies a primary instruction*** *referenced by a relocation table entry 260, as well as its shadow instruction,* ***into that relocation table entry 260***");

adding a jump instruction to the copied instructions to return to a next instruction after the original instructions (see at least col. 5:30-33 "*each primary instruction referenced by a relocation table entry 220-1 (with one exception noted below)* ***is replaced*** *by the native code loader 216* ***with a procedure call*** *to the dynamic resolver*"); and

replacing the original instructions while the original instructions are in the process of being executed on the processor with mark instructions and a transfer of control to a hook (see at least col. 5:30-35 "*each primary instruction referenced by a relocation table entry 220-1 (with one exception noted below) is replaced by the native code loader 216 with a procedure call to the dynamic resolver, and* ***its shadow instruction*** *(i.e. the next instruction after the primary instruction referenced by the relocation table entry)* ***is replaced by a NOP instruction***");

wherein the original instructions are part of the instruction set of the processor available to a user (see at least FIG. 3); and

wherein a number of times the mark instructions have been executed is countable, wherein the modified instructions include a resolver to determine a number of the instructions at a location of the original code that had already been executed (see at least col. 5:30-35 "*each primary instruction referenced by a relocation table entry 220-1 (with one exception noted below)* ***is replaced by the native code loader 216 with a procedure call to the dynamic resolver****, and its shadow instruction (i.e. the next instruction after the primary instruction referenced by the relocation table entry) is replaced by a NOP instruction*" – The instructions are stored in the relocation table are countable).

Claim Rejections - 35 USC § 103

9. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

10. Claims 1-3, 5, 7, 8, 10-22, 24-27, 29, 30, 32-38, 40-42, 44-50, 52, 53, 55-57, 59-64, and 67-73 are rejected under 35 U.S.C. 103(a) as being unpatentable over Mulchandani et al. (USPN 6,112,025), in view of McCormick, Jr. et al. (USPN 6,721,875).

As per claims 1, 27, 53, 67, 69, and 70:

Mulchandani teaches:

identifying original instructions to be changed while the original instructions are being executed on a processor (see at least col. 5:6-14 "*The relocation type of each patch instruction defines which bits of the instruction to be patched (i.e.* ***dynamically modified during execution of the program****) are provided by original object code instruction and which bits need to be replaced by the address of a constant pool entry. Furthermore, the object code array 252 in the initial compiled file 250 includes the object code for the instructions that will be* ***dynamically modified during program executing***");

copying the original instructions to a storage location (see at least col. 5:18-52 "*the native method loader 216* ***copies a primary instruction*** *referenced*

by a relocation table entry 260, as well as its shadow instruction, ***into that relocation table entry 260***");

adding a jump instruction to the copied instructions to return to a next instruction after the original instruction (see at least col. 5:30-33 "*each primary instruction referenced by a relocation table entry 220-1 (with one exception noted below)* ***is replaced*** *by the native code loader 216* ***with a procedure call*** *to the dynamic resolver*"); and

replacing the original instructions while the original instructions are in the process of being executed on the processor with mark instructions and a transfer of control to a hook (see at least col. 5:30-35 "*each primary instruction referenced by a relocation table entry 220-1 (with one exception noted below) is replaced by the native code loader 216 with a procedure call to the dynamic resolver, and* ***its shadow instruction*** *(i.e. the next instruction after the primary instruction referenced by the relocation table entry)* ***is replaced by a NOP instruction***");

wherein the original instructions are part of the instruction set of the processor available to a user (see at least FIG. 3); and

wherein a number of times the mark instructions have been executed is countable to determine a location of the processor's instruction pointer where execution should resume, in the patched instruction (see at least FIG. 3; see also col. 5:24-29 "*the* ***relative program counter values*** *in the relocation table entries are replaced during loading of the relocation table with the absolute program*

counter values for the corresponding instructions by ***adding the base address of the loaded native code to all the relative program counter values initially stored in the relocation table***" – The instructions are stored in the relocation table and therefore they are countable).

However, Mulchandani does not explicitly teach:

using atomic writes that guarantee that a result of the operation can be observed as compiled or not observed at all.

However, McCormick Jr. teaches:

using atomic writes that guarantee that a result of the operation can be observed as compiled or not observed at all (see at least 28:64-65 "*a pseudo-* ***atomic write*** *sequence must be used*").

Mulchandani and McCormick Jr. are analogous art because they are from the same field of endeavor. They all relate to patching software on-the-fly.

Therefore, it would have been obvious to one having an ordinary skill in the art at the time the invention was made to modify Mulchandani's approach to combine with McCormick Jr. One would have been motivated to modify in order to avoid any possibility of having a thread reading a partially updated bundle, the instruction bundle at the patch point, the start of an optimized procedure, or the start of a trace needs to be modified atomically.

<u>As per claims 2 and 41:</u>

Mulchandani in combination with McComick Jr. teaches all the limitations of the base claim as outlined above. Mulchanadani further teaches:

prior to the copying step, allowing a write operation on a page in memory where the original code is located (see at least col. 5:18-52 "*the native method loader 216* ***copies a primary instruction*** *referenced by a relocation table entry 260, as well as its shadow instruction,* ***into that relocation table entry 260***" – a write operation must be allowed in order to perform the copy operation).

As per claim 3:

Mulchandani in combination with McComick Jr. teaches all the limitations of the base claim as outlined above. McComick Jr. further teaches:

prior to the copying step, masking interrupts (see at least col. 29:20-25 "***Bundle patching must be thread safe****. If a thread is suspended on a certain syllable of a bundle which is subsequently patched by a dynamic instrumentation or post-link time optimizer tool,* ***the tool needs to make sure that the suspended thread can resume correctly on any syllable in the patched bundle***").

As per claim 5:

Mulchandani in combination with McComick Jr. teaches all the limitations of the base claim as outlined above. McComick Jr. further teaches:

after the replacing step, unmasking interrupts (see at least col. 29:20-25 "***Bundle patching must be thread safe***. *If a thread is suspended on a certain syllable of a bundle which is subsequently patched by a dynamic instrumentation or post-link time optimizer tool,* ***the tool needs to make sure that the suspended thread can resume correctly on any syllable in the patched bundle***" – resume the suspended can be considered as unmasking).

As per claims 7, 29, and 55:

Mulchandani in combination with McComick Jr. teaches all the limitations of the base claim as outlined above. Mulchanadani further teaches

wherein the mark instructions are the same length, in bytes, as the instructions to be patched (see at least col. 5:42-46 "*The reason that the* ***shadow instruction is replaced with a NOP instruction is as follows***. *In SPARC computers,* ***the shadow instruction*** *for every call instruction* ***is called a "delay slot" instruction****, and that delay slot instruction must actually be executed before the call instruction is executed*" - NOP instruction is also call delay slot instruction, therefore, the NOP instruction and the shadow instruction must be the same length).

As per claims 8, 30, and 56:

Mulchandani in combination with McComick Jr. teaches all the limitations of the base claim as outlined above. Mulchandani in combination with McComick Jr. further teaches

wherein the mark instructions are shorter in length, in bytes, as the instructions to be replaced (see McComick Jr. at least col. 7:43-45 "*patching technique which uses* ***long IP-relative branch instructions*** *to statically or* ***dynamically patch program code***"), and include NOP (no operation) filler (see Mulchandani at least col. 5:30-35 "*each primary instruction referenced by a relocation table entry 220-1 (with one exception noted below) is replaced by the native code loader 216 with a procedure call to the dynamic resolver, and its shadow instruction (i.e. the next instruction after the primary instruction referenced by the relocation table entry)* ***is replaced by a NOP instruction***").

As per claims 10, 32, 45, and 57:

Mulchandani in combination with McComick Jr. teaches all the limitations of the base claim as outlined above. Mulchandani further teaches:

wherein the resolver determines a number of instructions that had already been executed using the mark instruction (*instructions are stored in a relocation table and therefore they are countable*).

As per claims 11, 33, 46, and 59:

Mulchandani in combination with McComick Jr. teaches all the limitations of the base claim as outlined above. Mulchandani further teaches:

wherein, if the number of instructions that had already been executed is less than a number of original instructions to be changed, the resolver calls the copied instructions at the storage location so as to imitate a "no path installed" scenario (see at least col. 9:29-33 "*if procedure calls occupy five words, and the program counter for the present relocation table entry is less than five words away from the program counter for the preceding relocation table entry then step 310' is* ***skipped***" – no patching is performed).

As per claims 12, 34, 47, and 60:

Mulchandani in combination with McComick Jr. teaches all the limitations of the base claim as outlined above. Mulchandani further teaches:

wherein, after execution of the instructions at the storage location, the resolver returns control to the next instruction (see FIG. 7 - *The control resumes back to step 304*).

As per claims 13, 52, and 61:

Mulchandani in combination with McComick Jr. teaches all the limitations of the base claim as outlined above. Mulchandani further teaches:

enabling the functionality of the copied in the instructions in the storage location (see at least col. 5: 4 "*dynamically linked*").

As per claims 14, 15, 35, and 62:

Mulchandani in combination with McComick Jr. teaches all the limitations of the base claim as outlined above. Mulchandani further teaches:

wherein the enabling step comprising reconciling/alignment addressing in the instructions at the storage location (see at least col. 5:24-29 "*the relative program counter values in the relocation table entries are replaced during loading of the relocation table with the absolute program counter values for the corresponding instructions by* ***adding the base address of the loaded native code to all the relative program counter values initially stored in the relocation table***").

As per claims 16, 36, 48, and 63:

Mulchandani in combination with McComick Jr. teaches all the limitations of the base claim as outlined above. Mulchandani further teaches:

verifying that the original code is susceptible to patching (see at least col. 9:29-33 "*if procedure calls occupy five words, and the program counter for the present relocation table entry is less than five words away from the program counter for the preceding relocation table entry then step 310' is skipped*" – determine if patching is needed to perform).

As per claims 17, 37, 49, and 64:

Mulchandani in combination with McComick Jr. teaches all the limitations of the base claim as outlined above. Mulchandani further teaches:

wherein the verifying step determines whether any mark instructions are already in the original instructions (see at least col. 6:4-9 "*The NOP instruction is simply replaced by the original shadow instruction, unless the shadow instruction itself contains a reference to an external object component. In that case, the NOP instruction replaced by a call to the dynamic resolver and its shadow instructions is replaced by a NOP instruction*" – A determination must be perform to see if the NOP instruction existed in the modified instructions).

As per claims 18 and 50:

Mulchandani in combination with McComick Jr. teaches all the limitations of the base claim as outlined above. Mulchandani further teaches:

wherein the verifying step determines whether any copy protect instructions are already present in the original instructions (*The copy protection must be allowed in order to copy the instructions*).

As per claim 19:

Mulchandani in combination with McComick Jr. teaches all the limitations of the base claim as outlined above. Mulchandani further teaches:

wherein the verifying step determines whether the original instructions include a suitable jump point that can be modified to the transfer of control to the hook (see at least col. 4:57-67 "*in the object code file 250 generated by the Java*

to native code compiler 214 includes an object code array 252 and a relocation table 254. When the object code file 250 is initially generated, the relocation table 254 includes ***one entry 260 for each instruction of the compiled program*** *that references an object component*").

As per claim 20:

Mulchandani in combination with McComick Jr. teaches all the limitations of the base claim as outlined above. Mulchandani further teaches:

wherein the verifying step determines whether the original instructions represent valid instructions (*instructions must be valid instructions in order to perform patching*).

As per claims 21 and 38:

Mulchandani in combination with McComick Jr. teaches all the limitations of the base claim as outlined above. Mulchandani further teaches:

placing the hook in memory (see at least FIG. 2).

As per claim 22:

Mulchandani in combination with McComick Jr. teaches all the limitations of the base claim as outlined above. Mulchandani further teaches:

wherein the hook has been previously placed in memory (see at least FIG. 2).

As per claim 24:

Mulchandani in combination with McComick Jr. teaches all the limitations of the base claim as outlined above. Mulchandani further teaches:

wherein the atomic write replaces one instruction at a time (see at least col. 5:30-33 "***each primary instruction*** *referenced by a relocation table entry 220-1 (with one exception noted below)* ***is replaced by a native code loader 216 with a procedure call to the dynamic resolver)***" – In other words, replace one instruction at a time).

As per claim 25:

Mulchandani in combination with McComick Jr. teaches all the limitations of the base claim as outlined above. McComick Jr. further teaches:

wherein the atomic write replaces multiple instructions at a time (see at least col. 28:55 "*Instruction B****undle Patching***").

As per claim 26:

Mulchandani in combination with McComick Jr. teaches all the limitations of the base claim as outlined above. McComick Jr. further teaches:

wherein, for Intel IA-32 architecture, the atomic write uses any of "xchg", "lock cmpxchg8b", "lock cmpxchg", and "lock xchg" instructions (see at least col.

7:53 "*IA-64 Software Architecture*" – any one of these instructions can be used by Intel IA-64 architecture).

As per claims 40, 68, and 71:

Mulchandani teaches:

identifying original instructions to be changed while the original instructions are being executed on a processor (see at least col. 5:6-14 "*The relocation type of each patch instruction defines which bits of the instruction to be patched (i.e.* ***dynamically modified during execution of the program****) are provided by original object code instruction and which bits need to be replaced by the address of a constant pool entry. Furthermore, the object code array 252 in the initial compiled file 250 includes the object code for the instructions that will be* ***dynamically modified during program executing***");

copying the original instructions to a storage location (see at least col. 5:18-52 "*the native method loader 216* ***copies a primary instruction*** *referenced by a relocation table entry 260, as well as its shadow instruction,* ***into that relocation table entry 260***");

replacing the original instructions while the original instructions are in the process of being executed on the processor with mark instructions and a transfer of control to a hook (see at least col. 5:30-35 "*each primary instruction referenced by a relocation table entry 220-1 (with one exception noted below) is replaced by the native code loader 216 with a procedure call to the dynamic*

resolver, and ***its shadow instruction*** *(i.e. the next instruction after the primary instruction referenced by the relocation table entry)* ***is replaced by a NOP instruction***");

wherein the original instructions are part of the instruction set of the processor available to a user (see at least FIG. 3); and

wherein a number of times the mark instructions have been executed is countable to determine a location of the processor's instruction pointer where execution should resume, in the patched instruction (see at least FIG. 3; see also col. 5:24-29 "*the* ***relative program counter values*** *in the relocation table entries are replaced during loading of the relocation table with the absolute program counter values for the corresponding instructions by* ***adding the base address of the loaded native code to all the relative program counter values initially stored in the relocation table***" – The instructions are stored in the relocation table and therefore they are countable).

However, Mulchandani does not explicitly teach:

using atomic writes that guarantee that a result of the operation can be observed as compiled or not observed at all.

However, McCormick Jr. teaches:

using atomic writes that guarantee that a result of the operation can be observed as compiled or not observed at all (see at least 28:64-65 "*a pseudo-* ***atomic write*** *sequence must be used*").

Mulchandani and McCormick Jr. are analogous art because they are from the same field of endeavor. They all relate to patching software on-the-fly.

Therefore, it would have been obvious to one having an ordinary skill in the art at the time the invention was made to modify Mulchandani's approach to combine with McCormick Jr. One would have been motivated to modify in order to avoid any possibility of having a thread reading a partially updated bundle, the instruction bundle at the patch point, the start of an optimized procedure, or the start of a trace needs to be modified atomically.

As per claim 42:

Mulchandani in combination with McComick Jr. teaches all the limitations of the base claim as outlined above. Mulchandani further teaches:

> adding a jump instruction to the copied instructions to return control to a next instruction after the original instructions (see at least col. 5:30-33 "*each primary instruction referenced by a relocation table entry 220-1 (with one exception noted below)* ***is replaced*** *by the native code loader 216* ***with a procedure call*** *to the dynamic resolver*").

As per claim 44:

Mulchandani in combination with McComick Jr. teaches all the limitations of the base claim as outlined above. Mulchandani further teaches:

wherein the modified instructions include a resolver to determine a number of the instructions at a location of the original instructions that had already been executed (see at least col. 5:30-35 "*each primary instruction referenced by a relocation table entry 220-1 (with one exception noted below)* ***is replaced by the native code loader 216 with a procedure call to the dynamic resolver****, and its shadow instruction (i.e. the next instruction after the primary instruction referenced by the relocation table entry) is replaced by a NOP instruction*" – The instructions are stored in the relocation table are countable).

As per claims 72 and 73:

Mulchandani in combination with McComick Jr. teaches all the limitations of the base claim as outlined above. Mulchandani further teaches:

wherein the process of execution of the original instructions is not interrupted throughout the patching process (see at least col. 5:7-8 "*dynamically modified during execution of the program*" – patching while the program is being executed).

11. Claim 4 is rejected under 35 U.S.C. 103(a) as being unpatentable over Mulchandani et al. (USPN 6,112,025), in view of McCormick, Jr. et al. (USPN 6,721,875), and further in view of Scott et al. (USPN 6,615,329).

As per claim 4:

Mulchandani in combination with McComick Jr. teaches all the limitations of the base claim as outline above but Neither Mulchandani nor McComick Jr. teaches:

after the replacing step, disallowing a write operation on the page in memory where the block of code is located.

However, Scott teaches:

after the replacing step, disallowing a write operation on the page in memory where the block of code is located (see at least col. 9:53-54 "***disable write operation*** *to the protected area*").

Therefore, it would have been obvious to one having an ordinary skill in the art at the time the invention was made to modify Mulchandani and McComick Jr. to include the teaching of Scott. One would have been motivated to modify in order to protect the memory area by disallow or disable a write protection after data have been copied from memory to storage location.

12. Claims 6, 28, 43, and 54 are rejected under 35 U.S.C. 103(a) as being unpatentable over Mulchandani et al. (USPN 6,112,025), in view of McCormick, Jr. et al. (USPN 6,721,875), and further in view of Duesterwald et al. (USPN 6,928,536).

<u>As per claims 6, 28, 43, and 54:</u>

Mulchandani in combination with McComick Jr. teaches all the limitations of the base claim as outline above but Neither Mulchandani nor McComick Jr. teaches:

wherein the original instructions are changed in reverse order.

However, Duesterwald teaches:

> wherein the original instructions are changed in reverse order (see at least col. 9:50-52 "*steps my be executed out of order from that shown or discussed, including substantially concurrently or* ***in reverse order****, depending on the functionality involved*").

Therefore, it would have been obvious to one having an ordinary skill in the art at the time the invention was made to modify Mulchandani and McComick Jr. to include the teaching of Duesterwald. One would have been motivated to modify order to optimize their execution.

Conclusion

13. Applicant's amendment necessitated the new ground(s) of rejection presented in this Office action. Accordingly, **THIS ACTION IS MADE FINAL**. See MPEP § 706.07(a). Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of

the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the date of this final action.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Phillip H. Nguyen whose telephone number is (571) 270-1070. The examiner can normally be reached on Monday - Thursday 10:00 AM - 3:00 PM EST.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Wei Y. Zhen can be reached on (571) 272-3708. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

PN
5/7/2008

/Wei Zhen/
Supervisory Patent Examiner, Art Unit 2191